

receiving a write request that is to overwrite an existing data block in the mass storage device with a new data block;

executing the write request, such that the existing data block stored in the mass storage device is overwritten with the new data block.

2. A method as recited in claim 1, wherein the preservation memory comprises a volatile memory device.

3. A method as recited in claim 1, wherein the preservation memory comprises a portion of the mass storage device.

4. A method as recited in claim 1, further comprising the act of creating a backup copy of the plurality of data blocks stored by the mass storage device prior to the act of receiving the write request.

5. A method as recited in claim 6, wherein the act of creating the backup copy is conducted when the plurality of data blocks represent logically consistent data.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24

6. A method as recited in claim 1, further comprising the act of restoring the plurality of data blocks stored at the mass storage device to a state in which the plurality of data blocks existed at a previous point in time using the copy of the existing data block stored by the preservation memory.

7. A method as recited in claim 1, further comprising the act of storing a sequence of existing data blocks in the preservation memory in response to one or more write requests that are to overwrite the existing data blocks at the mass storage device and associating a time stamp with each of the existing data blocks of the sequence.

8. A method as recited in claim 7, further comprising the act of restoring the plurality of data blocks stored at the mass storage device to a state in which the plurality of data blocks existed at a previous point in time using the sequence of existing data blocks stored in the preservation memory.

9. A method as recited in claim 8, further comprising the act of experiencing a data corruption event in the mass storage device, wherein the act of restoring the plurality of data blocks is conducted to restore the plurality of data blocks to obtain non-corrupted data.

10. A method as recited in claim 8, wherein the act of restoring comprises the act of applying the data blocks of the sequence of existing data blocks stored in the

1 preservation memory to a current version of the plurality of data blocks stored at the mass
2 storage device in reverse chronological order.

3
4 11. A method as recited in claim 9, wherein the act of restoring the plurality of
5 data blocks is conducted to restore the plurality of data blocks to obtain non-corrupted
6 data and wherein the act of applying the data blocks comprises the act of determining
7 when the non-corrupted data is obtained.

8
9 12. A method as recited in claim 1, further comprising the act of establishing a
10 virtual device at the computer system that appears as if it contained the plurality of data
11 blocks stored at the mass storage device in a state in which the plurality of data blocks
12 existed at a previous point in time.

13
14 13. A method as recited in claim 12, further comprising the acts of:
15 receiving, at the virtual machine, a read request specifying a requested
16 data block; and

17 in response to the read request:

18 if the preservation memory includes a copy of a non-corrupted
19 version of the requested data block that was stored in the preservation
20 memory at or after the previous point in time, processing the read request
21 using the oldest such non-corrupted version of the requested data block
22 stored in the preservation memory; and

23 if the preservation memory does not include said copy of a non-
24 corrupted version of the requested data block that was stored in the

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24

preservation memory at or after the previous point in time, processing the
read request using a version of the requested data block stored in the mass
storage device.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24

14. In a computer system that includes a mass storage device that stores a plurality of data blocks, a method of restoring the data blocks of the mass storage device to a previous state, comprising the acts of:

iteratively performing the acts of:

receiving a write request that is to overwrite an existing data block in the mass storage device with a new data block;

prior to executing the write request, storing a copy of the existing data block in a preservation memory associated with the computer system and associating a time stamp with the copy of the data block in the preservation memory; and

executing the write request, such that the existing data block stored in the mass storage device is overwritten with the new data block;

experiencing a data corruption event whereby at least one of the plurality of data blocks stored in the mass storage device becomes corrupted; and

restoring the data blocks of the mass storage device to a previous state by applying the copies of the existing data blocks from the preservation memory to the plurality of data blocks of the mass storage device until the previous state is obtained.

15. A method as recited in claim 14, wherein the reverse chronological order is identified using the time stamps associated with the copies of the existing data blocks.

1 16. A method as recited in claim 14, further comprising the act of creating a
2 backup copy of the plurality of data blocks stored by the mass storage device prior to
3 iteratively performing the act of receiving a write request.
4

5 17. A method as recited in claim 16, wherein the act of restoring the data
6 blocks is conducted such that the previous state represents non-corrupted data and is
7 more recent than the backup copy.
8

9 18. A method as recited in claim 14, wherein the iteratively performed act of
10 storing a copy of the existing data block is performed on the data block level and
11 independent of any file structure associated with the mass storage device.
12

13 19. A method as recited in claim 18, wherein the plurality of data blocks
14 represent data stored in specified sectors of the mass storage device.
15
16
17
18
19
20
21
22
23
24

1 20. In a computer system that includes a mass storage device that stores a
2 plurality of data blocks, a method of establishing a virtual device that enables access to
3 the plurality of data blocks as the data blocks existed at the mass storage device at a
4 previous point in time, comprising the acts of:

5 storing a current version of the plurality of data blocks in the mass storage
6 device;

7 maintaining, at a preservation memory associated with the computer
8 system, copies of previous versions of data blocks that have since been
9 overwritten at the mass storage device in response to write requests, the previous
10 versions of the data blocks being associated with time stamps specifying a
11 chronological order in which the data blocks were overwritten;

12 providing access to the current version and the copies of the previous
13 version through a virtual device, wherein the virtual device, in response to read
14 request specifying a particular data block as it existed at the previous point in
15 time, determines whether to access the current version of the data block from the
16 mass storage device or the previous version of the data block from the
17 preservation memory.

18
19 21. A method as recited in claim 20, wherein:

20 the current version of the plurality of data blocks includes one or more
21 corrupted data blocks; and

22 at the previous point of time, the plurality of data blocks were non-
23 corrupted.
24

1 22. A method as recited in claim 20, wherein, in response to the read request,
2 the virtual device performs the acts of:

3 if the preservation memory includes a copy of a non-corrupted version of
4 the requested data block that was stored in the preservation memory at or after the
5 previous point in time, processing the read request using the oldest such non-
6 corrupted version of the requested data block stored in the preservation memory;
7 and

8 if the preservation memory does not include said copy of a non-corrupted
9 version of the requested data block that was stored in the preservation memory at
10 or after the previous point in time, processing the read request using a version of
11 the requested data block stored in the mass storage device.

12
13 23. A method as recited in claim 20, further comprising experiencing a data
14 corruption event whereby one or more current version of a data block represents
15 corrupted data, wherein the virtual device providing access to the plurality of data blocks
16 as the data blocks existed at the mass storage device at a previous point in time in a non-
17 corrupted state.

18
19 24. A method as recited in claim 20, wherein the preservation memory
20 comprises a portion of the mass storage device.

1 25. A computer system that provides access to data blocks as the data blocks
2 existed at a previous point in time, comprising:

3 a mass storage device that stores a plurality of data blocks that can be
4 overwritten in response to a write request;

5 a preservation memory that receives and stores a copies of the data blocks
6 that are to be overwritten prior to the write request being processed and that
7 associates a time stamp with each said copy of a data block; and

8 a virtual device that provides access to a current version of the data blocks
9 stored at the mass storage device and the copies of the data blocks at the
10 preservation memory, wherein the virtual device, in response to read request
11 specifying a particular data block as it existed at the previous point in time,
12 determines whether to access a current version of the specified data block from
13 the mass storage device or a copy of the specified data block from the
14 preservation memory.

15
16 26. A computer system as recited in claim 25, wherein the preservation
17 memory comprises a portion of the mass storage device.

18
19 27. A computer system as recited in claim 25, wherein the virtual device, from
20 the standpoint of a data access program, appears as if it contained the data blocks as the
21 data blocks existed in the mass storage device at the previous point in time.

22
23 28. A computer system as recited in claim 25, wherein the virtual device, in
24 response to the read request, performs the acts of:

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24

if the preservation memory includes a copy of a non-corrupted version of the specified data block that was stored in the preservation memory at or after the previous point in time, processing the read request using the oldest such non-corrupted version of the specified data block stored in the preservation memory; and

if the preservation memory does not include said copy of a non-corrupted version of the specified data block that was stored in the preservation memory at or after the previous point in time, processing the read request using a version of the specified data block stored in the mass storage device.

29. A computer system as recited in claim 26, further comprising means for creating a backup copy of the data prior to receiving the write request.